Flower Recognition Using Convolutional Neural Network

Burak Erdil Biçer*, Javad Ibrahimli**, Mehmet Anıl Korkmaz§
bicerb20@itu.edu.tr*, ibrahimli21@itu.edu.tr**, korkmazm19@itu.edu.tr§
*** §Electronics and Communications Engineering, Istanbul Technical University, Istanbul, Turkey

***Abstract*—Artificial intelligence is the new frontier in the history of technological development, opening the way to an absolutely new phase with qualitative changes in the most diverse industries. One of the game-changing technologies is Convolutional Neural Networks (CNNs), which have shown good results in various tasks related to image recognition. In this paper, the application of CNN in the domain of flower recognition, which has large implications for agriculture and marketing, is presented.**

***Index Terms*—Artificial Intelligence, Convolutional Neural Networks, Image Recognition, Flower Classification, Precision Agriculture, Deep Learning, Data Augmentation, MobileNetV2.**

I. Introduction

With the rise of artificial intelligence over time, a completely new chapter of technological growth has opened. The manner in which we interface with the world has now changed altogether. A very important part of these marvelous strides is a sub-area of artificial intelligence called convolutional neural networks, abridged to CNN.

Convolutional neural networks are one type of machine learning algorithms that can be used to infer deep learning and to realize visual recognition. It can apply to image recognition and classification, which is perfect for work such as flower recognition. CNNs are inspired by the hierarchical processing of human vision.

A CNN usually consists of several layers: convolutional layer, pooling layer, and fully connected layers. Conclusively, the convolutional layers are the very basic building blocks through which the input image goes, given a set of filters typically called kernels. A set of these filters will slide over the image, identifying different types of features such as edges, textures, or patterns. This is a set of operations, the end result of which is a set of feature maps highlighting particular features present in different regions of the image.

Pooling layers follow convolutional layers to reduce the spatial dimensions of the feature maps. And the common operations include max pooling, in which the maximum value of every patch of the feature map is taken, and average pooling, in which the average value is computed for that.

These are typically tacked at the end of the network and extract high-level features derived from the convolutional and pooling layers for the purpose of classification. The final layer provides the probabilities of the image belonging to different classes, hence getting the final decisions. A CNN contains inside it, and training bases its way upon backpropagation. That is, the network predictions are paired with the ground truth, and the errors in the predictions are fed back through the network structure. This kind of process goes on iteratively until the model's performance is acceptable. The capability of CNNs to learn and extract relevant features from raw image data has been one of the backbones for several state of the art flower detection models. Visual explanation of the CNN is given in the Fig. 1.

In the blossoming field of computer vision, the identification and classification of different types of flowers by analyzing images is of the essence [1], [2]. This has been a good course to prove the growth in artificial intelligence. This project report outlines research with the use of CNN for accurate classification of flower species from digital images. With the power of deep learning, our program not only distinguishes between a bunch of flower types but also discloses the functioning of such neural network-based image recognition.

The following sections of this report will take the readers through an in-depth literature review, explaining the state of the art; our work, which will detail the motivation, dataset, data preparation, exploratory data analysis, and prediction methodologies used; followed by a candid discussion on the challenges we faced during the project's fruition. Lastly, we will meet the clarion call of future work, promising to further grow the field of flower recognition.

II. Literature Review

Hanafiah et al. (2022) studied deep CNN models that used a transfer learning approach to recognize flowers [3]. In their research, Hanafiah et al. implemented the application using a Kaggle benchmark dataset. They evaluated two of the most popular image classification models: AlexNet and VGG16. As shown, VGG16 performed slightly better than AlexNet, with an accuracy of 95.02% in comparison to 85.69%. The authors identified that the image type and layers of CNN most probably affect recognition performance. The study revealed the capacity of the CNN to recognize various species of flowers. The application of CNN in this study opened up a perspective in botanical study and the agricultural field.

Mete and Ensari (2019) introduced a classification system to hybrid flowers, where they integrated deep CNNs with many classifiers, including SVM, Random Forest, and KNN [4]. They applied a very large sum of machine learning classifiers such as SVM, Random Forest, KNN, and MLP. They empirically tested both the Oxford 17-Flowers and Oxford 102-Flowers augmented datasets. They were able to achieve

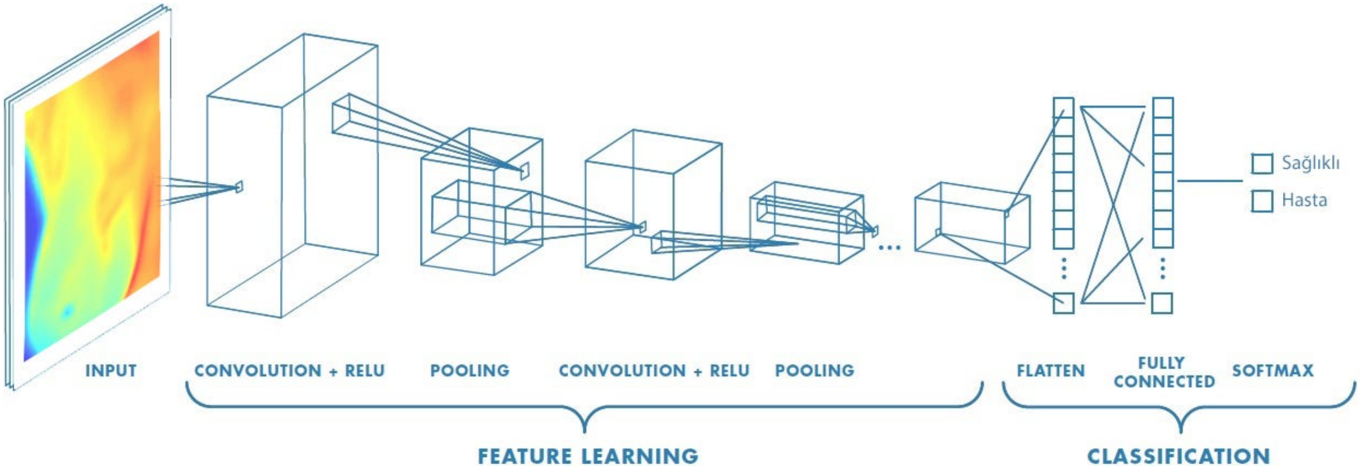

Fig. 1: Figure illustrates a convolutional neural network (CNN) used for image processing. It starts with an input image, followed by multiple layers of convolution and pooling. Convolution layers use filters to capture spatial features and are followed by ReLU functions to add non-linearity. Pooling layers reduce dimensionality to lower computation. The network's deeper layers refine feature extraction before a flattening step converts the data into a 1D vector. This vector feeds into fully connected layers that integrate these features. After, a softmax layer finally classifies the output into categories like 'Healthy' or 'Ill'. This architecture is effective for recognizing and differentiating complex patterns in images.

an outstanding 98.5% accuracy rate with the SVM classifier on the Oxford 102-Flowers dataset. In this work, the feature extraction of adding several classifiers in combination with CNN is to be shown for the improvement of the accuracy and robustness of the flower classification system, which really is very important for botany and agriculture. They worked on the classification of flowers using CNN-based architecture and transfer learning, where the VGG16, MobileNetV2 [5], and ResNet50 models were experimented with. This research showed that a maximum validation accuracy of 97.07% was registered with the ResNet50 model. The research showed that transfer learning trains a CNN much better than from the beginning, especially in cases that have only a small number of labeled images. This method has proven to be useful in agricultural and botanical practices because, in most cases, the datasets are not labeled. The paper confirms the relevance of transfer learning in harnessing pre-trained models for effective and accurate classification of flowers in works done by Narvekar and Rao (2020) [6].

In one research study by Yifei et al. (2022), an enhanced CNN architecture was proposed with respect to the classification of flower images based on the traditional CNN design for the purposes of enhancing accuracy and reducing computational complexity [7]. The superior performance of their model for classifying flower images draws attention to the creation of optimized CNN architectures for the purpose of achieving higher rates of accuracy. This, therefore, should relate to people who are interested in developing more efficient and precise systems based on CNN for flower recognition and emphasize on the need for refining the steps of CNN design and implementation by continuous improvement.

Rajkomar and Pudaruth (2023) integrated deep CNNs with traditional machine learning algorithms to come up with a robust system of flower classification [8]. The Oxford 17-Flowers and Oxford 102-Flowers datasets were taken, and as always, through this hybrid approach, high accuracy rates were retrieved. In further works, it has been shown that the deep learning-based feature extraction, along with machine learning classifiers for the final classification, can also be used to enhance the performance of the system greatly. This paper demonstrates that sometimes, hybrid approaches are very effective in overcoming the related challenges of the area of flower classification—for example, inter-class similarity and intra-class variations.

III. Our Work

A. Motivation

Detection of flowers, based on CNNs, is the field that leads research that will in the future implement up-to-date computer vision techniques in agriculture and marketing.

With the help of CNNs, flower detection is further evolving traditional farming processes in the agricultural setting. The advanced machinery, with the help of computer vision technology, is therefore capable of distinguishing between weeds and actual crops. This correct identification allows for targeted weed control, where it is possible to avoid mechanization of a large area. One of the consequent results is that the production of crops becomes very efficient and their yields improve as the cost of production becomes very low. The system of flower detection also facilitates monitoring crop health and the time of flowering in order to predict when each crop will flower, obviously important for maximum harvesting scheduling. From this basis, it can be observed that such technologies greatly reduce manual work and increase accuracy with the aim of inventing better, sustainable, and more profitable farming.

This has further advantages in terms of the possibility to monitor fields on a continuous basis with minimal human intervention. This constant surveillance will allow the first signs of disease or damage from pests to be detected, and preventive measures can be taken to avoid potential losses. Possible use case is shown in the Fig. 2.

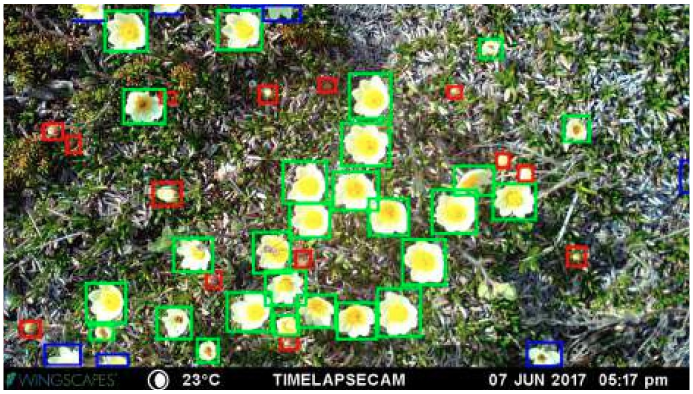

Fig. 2: Automatic detection of flowers in a meadow using object detection techniques. The boxes reflect the flowers detected using the color-coding with different colors. Green means that a flower is detected correctly, red represents a false positive, and blue is for an object that cannot be determined [9].

Thus, the real-time datas further maintain the environment and lessen cost in applying the fertilizers and pesticides in exact proportions. In other words, the ability to detect flowers through CNNs gives farmers additional tools to extend control over their fields better. This guarantees that resources applied are used the best and results in excellent quality crops. Visualized representation of the how system might work has shown in the Fig. 3.

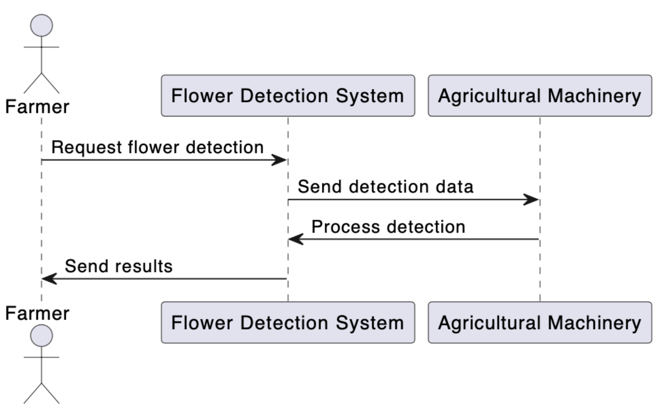

Fig. 3: This sequence diagram illustrates the interaction between a farmer, a flower detection system, and agricultural machinery.

Research in the use of flower detection for agricultural machinery and marketing spans various applications, primarily focusing on enhancing automation in floriculture and improving crop management. The paper by Shree and Kaur (2019) surveys flower detection techniques using deep neural networks, highlighting the challenges in automated flower harvesting, such as yield estimation and the separation of flowers from the background [10]. This study emphasizes the significant economic impact of floriculture in India, particularly for marigold production. Khanal et al. (2023) explore a machine vision system for early-stage apple flowers and clusters detection, which is crucial for precision thinning and pollination [11]. They employ the YOLOv5 algorithm, achieving an 81.9% mAP accuracy, showcasing the potential of robotics in orchard management to enhance fruit quality and profitability. Subramanian et al. (2022) introduce "Flower-Bot," a robotic system designed to automate flower picking, addressing labor challenges and improving efficiency through night vision cameras and Raspberry Pi processing [12]. These advancements demonstrate the integration of deep learning and robotics in floriculture, paving the way for more efficient and profitable agricultural practices.

In the marketing world, the use of CNNs for flower detection changes understanding and responsiveness to consumer preferences.

Apart from that, there are also algorithms over the internet that can scan images using CNN and identify the trends in the types of flowers. It, therefore, only identifies what is a trend with the masses of consumers at any one time. The greater interest of it all is that they will run a campaign for you where only that particular set of flowers will be shown in the ads, making the campaign so much more interesting and relevant to the target audience. Such a focused approach not only enhances consumer interest but also increases sales because the product offering is well geared towards what is happening at that moment. Flower detection technology can be integrated into an e-commerce platform for more personal shopping experience. For instance, application of the flower detection algorithms in suggesting bouquets that are in trend or further refined to take into account individual customer touchpoints. Social media platforms may enrich their content suggestions to ensure that users view posts and ads through which their favorite flowers will be included. They get very good loyalty with extremely high conversion rates. Furthermore, by dynamically altering themselves with the shift in preferences, marketing organizations can assuredly remain competitive within a continually changing market environment and landscape. Real-time data analysis allows a marketer to be agile toward changes in the customer's behavioral pattern and thus change strategies to meet emerging demands quickly. This agility is key for one to stay on course and derive the most from all marketing efforts. Application of flower detection using CNN is, therefore, promising in two fields: agriculture and marketing. More efficient and accurate use realized through innovation in computer vision that has growth potential in these industries consequently means more customer satisfaction. The current convergence of artificial intelligence with its practical application in these areas presents frontier potential for CNNs and gives way forward for further advancement concerning the producers and consumers. Mindmap of the concepts in the motivation can be seen in the Fig. 4.

B. Dataset

The data set applied for recognition of flowers in this project contains 4242 images of five disjointed flower types: tulip, daisy, dandelion, rose, and sunflower shown in the Fig. 5. This data set is called "Flowers Recognition" from Kaggle [13].

The dataset consists of tulips, daisies, dandelions, roses, and sunflowers, while in each category, many represent the variations and conditions of the flowers, such as angle, lighting, and background. Such variety shall be required to train and build a strong convolutional neural network, a network that might come to identify the class of flowers in any or all of these contexts or any of the environments with a high degree of assurance.

These images are obtained from various sources via scraping from the internet; some of the sources include Flickr, Google Images, and Yandex Images. This type of scraping makes sure that the image dataset is varied in one way or another. The diversity in the sources of images reflects the strong generalization of the CNN model in making associations of features with flowers in their contexts and environments. It will help ensure learning from a representative sample of data by adding more data sources, which ultimately helps to reduce the risk of overfitting to a particular style or type of imagery in the development of a robust and accurate flower recognition system.

This dataset is chosen for the project since, the dataset is pretty balanced since no single class will dominate the others. This balance is essential in having a model learn nicely without bias toward any specific class, as it ensures there is good generalization about robust performance across the different classes of flowers.

The three datasets of flowers used for comparison that include Flowers Recognition [13], 102 Category [14], and 17 Category [15]. Flowers Recognition is a dataset with five classes in .jpg format, with an approximate size of 236 MB. Moreover, the dataset does not include separated sets for validation and testing and contains some 1000 files available in every class. The 102 Category dataset is also in a .jpg format. The 102 Category dataset consists of 102 classes, with an approximate size of about 329 MB. It also lacks well-separated test/validation sets and contains 40–258 files per class. The 17 Category dataset has a 17 class file stored in the .jpg format and is about 57MB. Similar to other sets, it doesn't have well-separated tests and validation sets. However, it contains around 80 files per class. All these properties are given in the Table I.

All these characteristics make the Flowers Recognition dataset a good choice for this project. It is of manageable size thus processing and storage will be easy. The dataset's five classes are in the suitable range for the task to be done, so it is an ideal balance of neither complex nor easy. With approximately 1,000 files per class, the dataset also reflects a considerable amount of data per category, making the model robust at training time. This kind of volume for each class can consequently result in classification accuracy and generalization of the model into better representation in a class. Therefore, the Flowers Recognition dataset is quite relevant for this.

TABLE I: Comparison of Flower Datasets

Dataset	Flowers Recognition [13]	102 Category [14]	17 category [15]
Classes	5	102	17
File Format	.jpg	.jpg	.jpg
Size	˜236 Mb	˜329 Mb	˜57 Mb
Separated test/ver.	×	×	×
Files per Class	˜1000	˜40-258	˜80

C. Data Preparation

In this study, a dataset consisting of 4242 images of five types of flowers was used. In carrying out the steps of preparing the data, a number of important steps were administered to make the dataset ready for training using the MobileNetV2 model. The elaboration steps can be seen in the Fig. 6.

We loaded the dataset within the provided directory. In this step, we read every image and extracted corresponding class labels from the structure of the directory. Then file paths and labels are merged into a Pandas DataFrame for effective data manipulation. To allow the model generalization potential and prevent overfitting, several data augmentation techniques were carried out, namely: random rotation width and height shifting, shear, zooming, and horizontal flipping. This data augmentation was carried out within the ImageDataGenerator class from TensorFlow. The procedure of splitting into training and validation sets was performed. The training set is for modeling, and the validation is for checking the performance of the trained model. A stratified split of the data was used, maintaining the proportion of each class in both sets. We visualized class distribution and information on sample images from various classes in the dataset. We can use this information to understand the diversity and balance of the dataset.

These several steps organized the data prettily and augmented the data to raise variability, and proportionally splitting was done to train and validate the model. This very exhaustive preparation laid a good foundation for training the MobileNetV2 model on perfect recognition of different species of flowers.

D. Exploratory Data Analysis

EDA was carried out to gain insight into the distribution and characteristics of the recognition of the dataset used for flowers. The dataset consists of 4242 images of 5 different flower types. The following are the major steps which we applied in the process of EDA. Knowing class distribution and samples through visual inspection: In this regard, we do the plotting for a better understanding of the representation of the flower types. This work was done using a pie chart. A pie chart

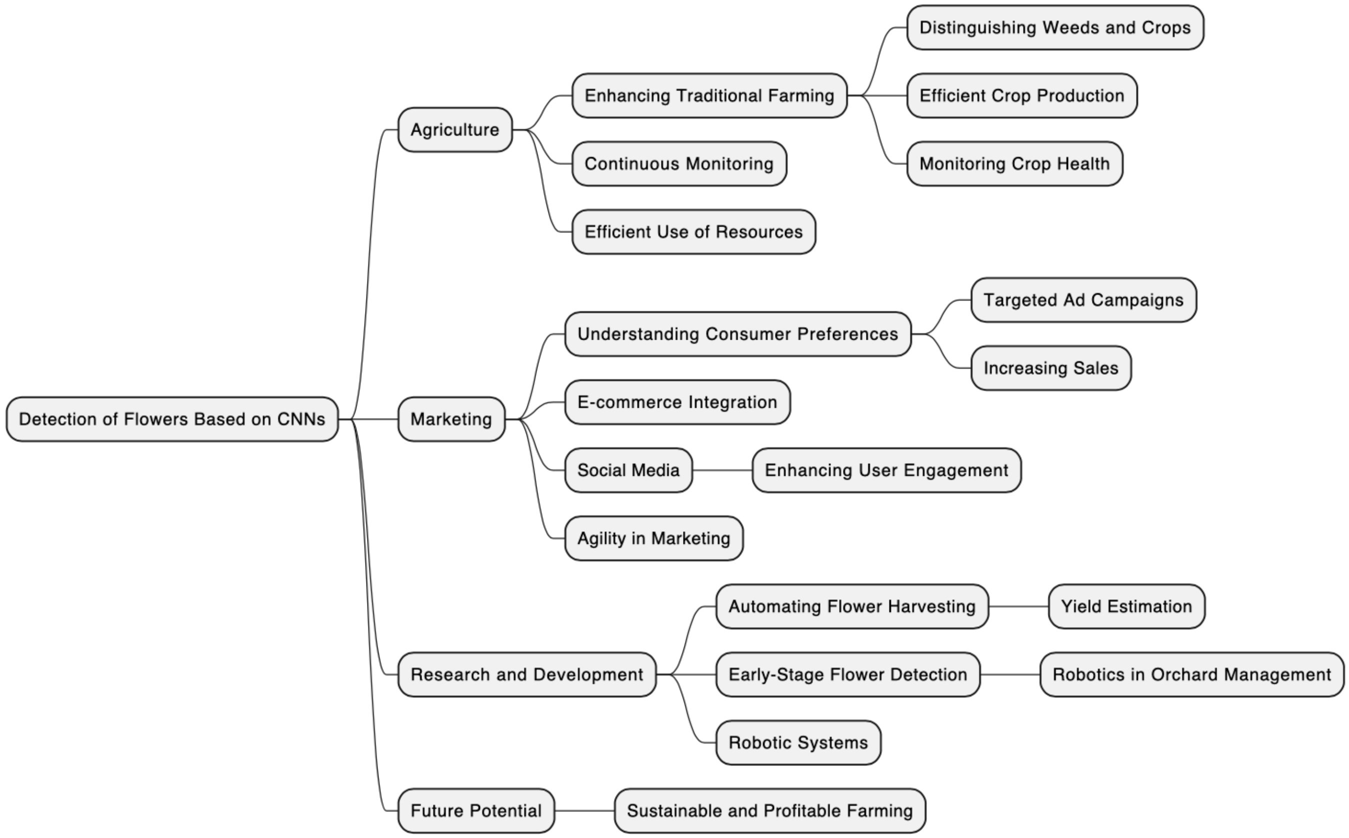

Fig. 4: Diagram illustrates the various aspects of flower detection based on CNNs. It organizes the concepts in the motivation subsection.

Fig. 5: The image of five different types of flowers used for a flower recognition project using CNN. From left to right, the flowers are dandelion, tulip, rose, daisy and sunflower. These images represent the diverse visual features used in training the convolutional neural network for accurate flower recognition.

can be an excellent visualization when one is trying to look at the imbalance in the number of samples in a dataset, which can be so crucial in terms of the performance of a model. This pie chart is displayed in Fig. 7.

From the distribution analysis, it is seen that the dataset is somewhat balanced, and no class seriously dominates the dataset. Such a balance is crucial in terms of having good generalization of the model when applied on this data set and not being biased toward a certain class.

After the numerical summaries, a sample of images from each class was first inspected. This will be helpful in verifying the quality and diversity of the images within each class. A few class samples are shown in Fig. 8. This way, quality assurance is given, ensuring that the dataset is available for use with different image types for each class, thus benefiting the model in terms of robustness. Therefore, the outcomes of the EDA provided key insights necessary to conduct the subsequent stages of data preparation and model training.

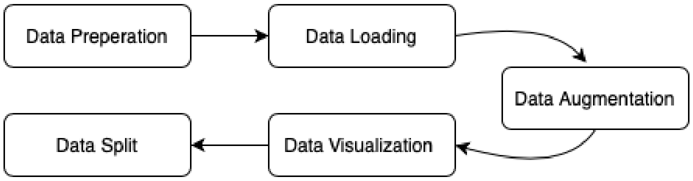

Fig. 6: Flowchart that outlines different steps in a data processing pipeline.

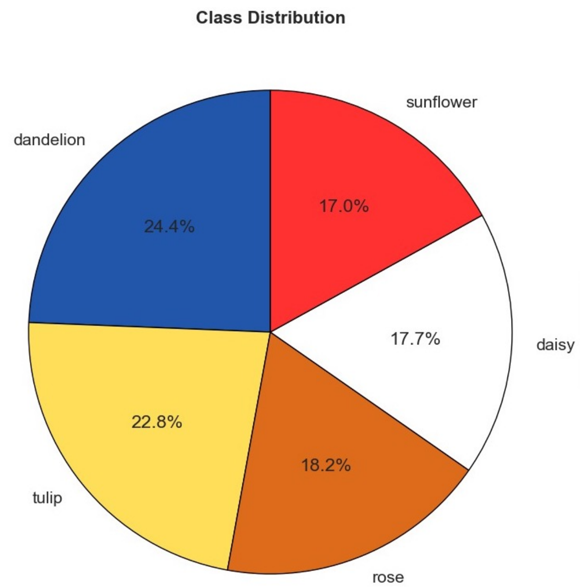

Fig. 7: The pie chart shows the proportion of each flower type in the dataset.

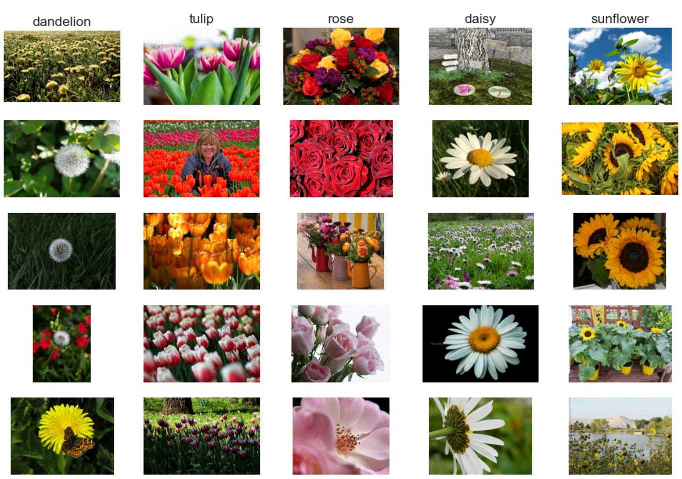

Fig. 8: Figure illustrates the variety of flowers present in the dataset, including dandelions, tulips, roses, daisies, and sunflowers. Each column represents a different flower type, showcasing multiple images to highlight the diversity within each category.

E. Creating Model

Several data preparation procedures have been applied to train the MobileNetV2 model for the purpose of recognition of flowers with high accuracy and robust performance. This can be seen in the following sub-sections: data preparation, model compilation, followed by the training and evaluation of the model.

1) Data Preparation: The dataset was first partitioned into a training and test set following a 90-10 partition. This will let a substantial part be in the training set that will assure intensive learning while splitting it into a set satisfactory to evaluate the generalization capabilities. Further, image augmentation of the training data was carried out. A number of techniques such as random rotations, shifts and flips, and brightness are applied in the images of the data set. How this techniques are applied is shown in the Fig. 9.

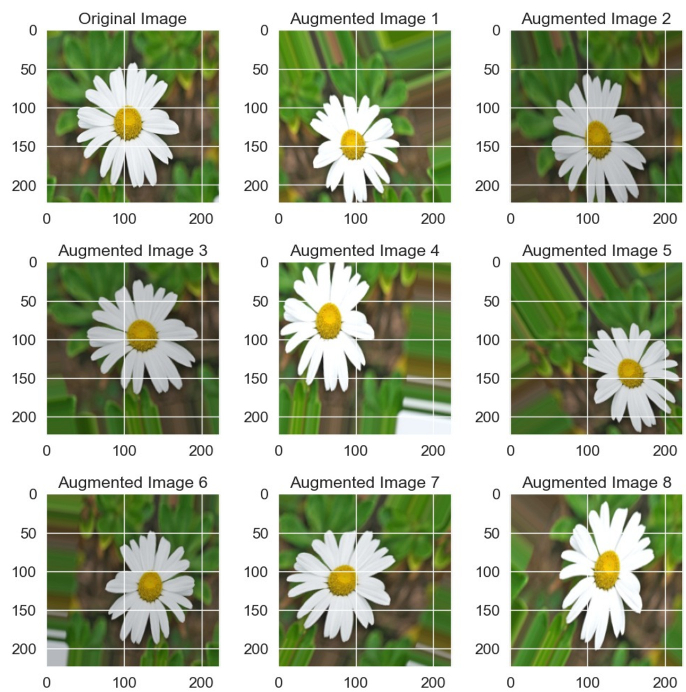

Fig. 9: Data augmentation applied to an image of a daisy using various techniques such as shifting, rotation, brightness adjustment and flips. The original image is shown in the top left, with augmented versions displayed in the remaining grid positions.

2) Model Compilation: The pre-trained MobileNetV2 model was downloaded from the server. The models are fine-tuned for the task of flower recognition. There was a change in the model architecture to accommodate the change for the number of flower classes in the dataset. Model compilation was done using Adam optimizer using a learning rate of 0.001. The loss function applied is the categorical cross-entropy, a good choice for multi-class classification tasks. Model fitting is done with accuracy and loss monitored in the training process.

3) Training Process: We applied early stopping to the model at around the 26th epoch. Early stopping is a regularization technique in machine learning that helps prevent overfitting. Early stopping is a way of regularization by which the performance on a validation set gets monitored during training; then, in the process, when the performance starts deteriorating, this training process is stopped. As a result, the model does not continue fitting to the noise of the training set; therefore, it will generalize better to unseen data, improving predictive performance on new datasets.

Model fitting to the training data was done using the Keras framework. In the callbacks for each epoch, the model checkpoint and early stopping were used to stop the bias from overfitting. We then allow it to run for 30 epochs at each time using a batch size of 32; if the particular epoch at that time does not outperform the model on the validation set, the training is stopped thereafter. However, training stopped effectively at the 26th epoch, because there were installed mechanisms for early stopping.

4) Evaluation: The model was trained, and evaluation on the test set was done. The evaluation set metrics include test loss and accuracy. Besides, confusion matrices and classification reports were generated to understand the detailed analysis of model performance over different flower classes.

5) Results Visualization: The training progress was visualized by plotting the accuracy and loss. They show how well the model learns during the training process. That further helps to find problems such as overfitting or underfitting. The predictions of the network can be qualitatively visualized by overlaying them on top of a subset of the test images. In general, except for the very deep architectures, the MobileNetV2 model can provide a very high level of accuracy, which proves its effectiveness on the task of flower recognition. Accuracy and loss graphs are given in the Fig. 10.

If the model were overfitting through training, the accuracy would still be very high, while the validation accuracy would be notably lower and could decrease at some epochs. Furthermore, the validation loss will always stay higher than the training loss. It may even increase, indicating that the model is not generalizing well but fitting too well to the training data. However, in the given graph, the close alignment between training and validation metrics suggests the model maintains a good balance and generalizes well to new data.

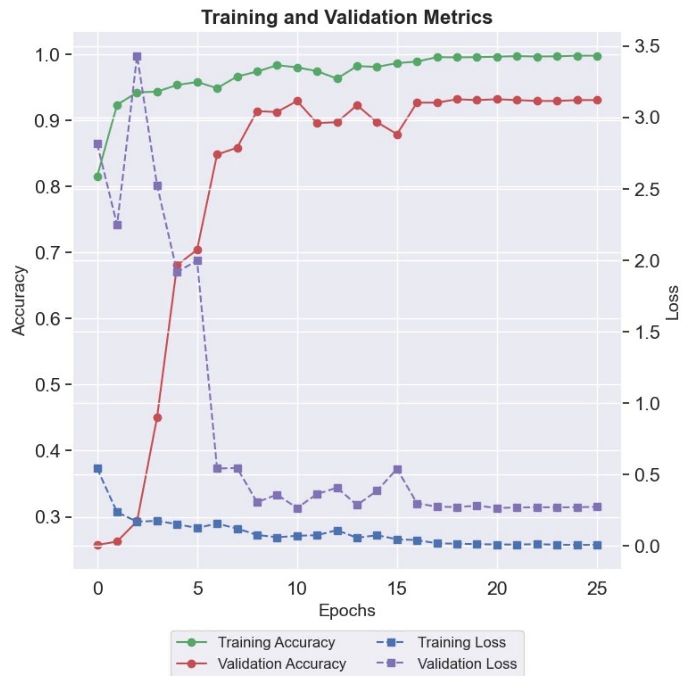

Fig. 10: Training and validation metrics over 26 epochs. The graph shows the training and validation accuracy (left y-axis) and loss (right y-axis) for each epoch. The training accuracy (green) increases steadily and stabilizes near 1.0, while the validation accuracy (purple) fluctuates initially but stabilizes around 0.9. Training loss (red) decreases sharply and stabilizes near 0.1, while validation loss (blue) shows fluctuations and stabilizes around 0.2.

F. Prediction

The prediction phase is the main part of our CNN flower recognition project. The intention of this stage is to measure how effective the model that has been trained will be in assigning the flower species for the input image of relevance. Following the exhaustive preparation and scrubbing through, the data training and validation of the model, the attestation is on how well the model could perform on the unseen data and how well it generalizes the set of data beyond the training set.

In the Fig. 11 classification results of randomly selected different flower images can be observed. The images represent true and predicted class labels associated with each flower image. The green border and wording represent a correct prediction, while the red border and wording represent an incorrect prediction. The model gets most of the flowers right, making only one misclassification whereby it classifies a dandelion as a sunflower. As it can be observed, model in the project can identify different flowers from one to another.

One of the stages of the predicting is processing stages. In the Fig. 12 processing stages of the CNN model on a dandelion image can be observed. The image demonstrates the different stages of image processing in our CNN model. Original image shows a dandelion image as obtained from the internet using the request library. Normalized image shows the same image after normalization to enhance contrast and color distribution. Resized image shows the normalized image resized to the input dimensions required by the CNN model for further processing and classification.

Table II presents the performance metrics of a classification model evaluated on a test dataset consisting of five classes which are daisy, dandelion, rose, sunflower, and tulip. The metrics included are precision, recall, F1-score, and support.

The class column indicates the type of flower being classified. Precision is defined as the ratio of correctly predicted positive observations to the total predicted positives. Mathematical expression of the precision is given in the Eq. 1.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

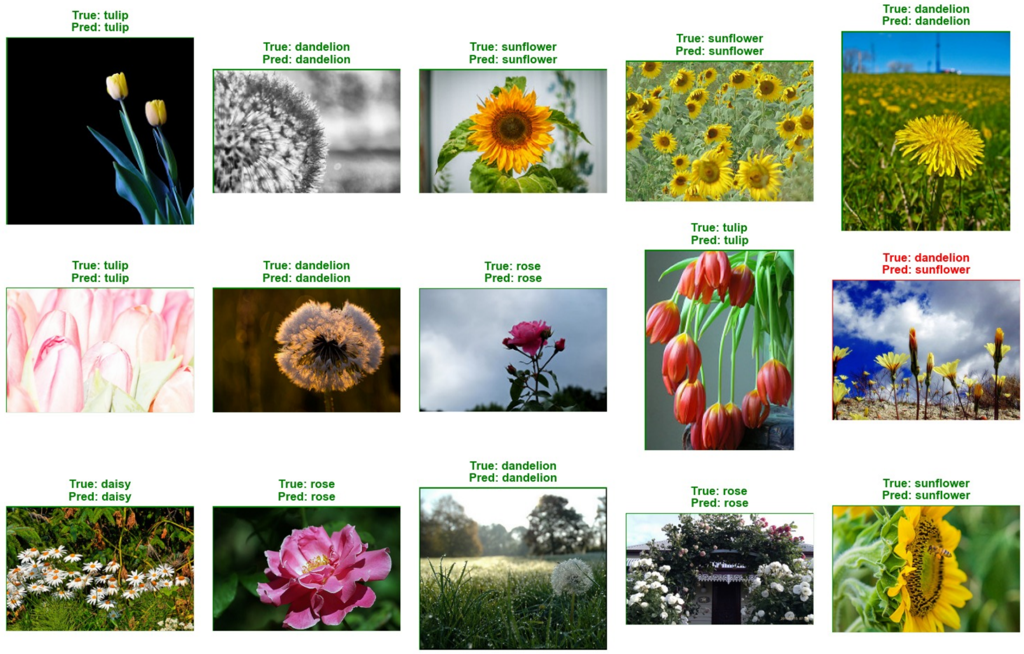

Fig. 11: Classification results of randomly selected different flower images.

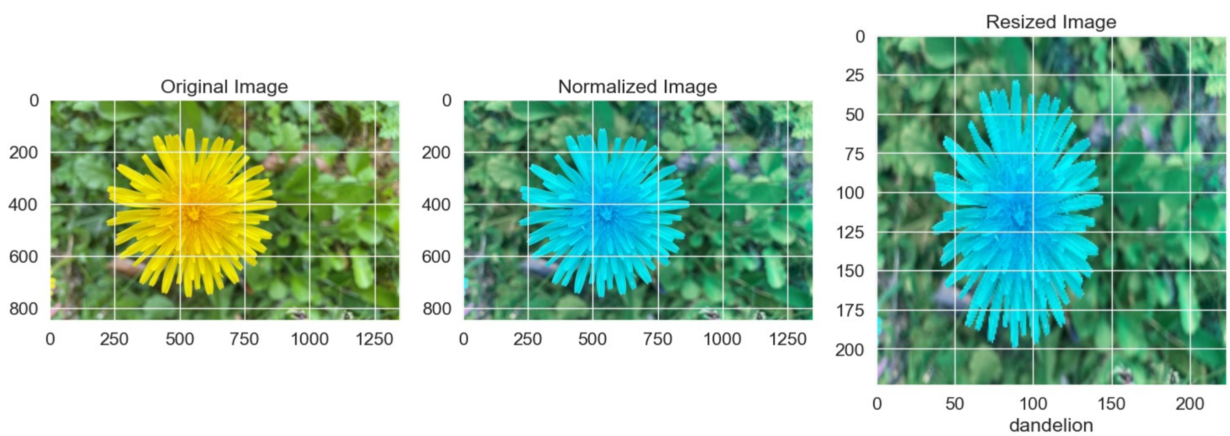

Fig. 12: Processing stages of the CNN model on a dandelion image.

Where TP represents true positives and FP represents false positives.

The recall is the ratio of correctly predicted positive observations to all the observations in the actual class. It is calculated as it shown in the Eq. 2.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Where FN represents false negatives.

The F1-Score is the weighted average of precision and recall. F1-Score is calculated as it is shown in the Eq. 3.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

This model's good performance is quite constant across classes, with very high values in precision, recall, and F1-scores. In fact, dandelions have the highest precision in the model. Its value is very high at 0.98, thereby ensuring very few false positives. Sunflowers have the highest F1 score, which is 0.96, representing a good balance of precision and recall. The general model accuracy and its ability to rightly classify an instance is 94%. Macro and weighted average representations give the best of the performance of a model owing to the potential imbalance among classes.

In the Fig. 13, confusion matrix depicting the performance of the model with respect to all classes of the flowers. Each row is where the actual flower belongs, and each column is the predicted class. These values of the matrix highlight the proportions of correct and incorrect predictions for each class. The high diagonals show high accuracy values on the model for the predictions that put most flowers into their respective

TABLE II: Classification Report

Class	Precision	Recall	F1-Score	Support
Daisy	0.97	0.94	0.95	77
Dandelion	0.98	0.93	0.95	95
Rose	0.90	0.96	0.93	89
Sunflower	0.96	0.96	0.96	67
Tulip	0.93	0.95	0.94	104
Accuracy	-	-	0.94	432
Macro Avg	0.95	0.94	0.95	432
Weighted Avg	0.95	0.94	0.94	432

classes. We used a normalized confusion matrix to make a detailed analysis, contrary to the standard confusion matrix. The normalized confusion matrix illustrated, as shown in the plot above, a clearer view on the ratio of predictions for each class. This visualization takes a clear view where the model does well and where the model doesn't.

Prediction is high with daisies, with both precision and recall at 94%. Some minor misclassifications, however, are available within the dandelions and the sunflowers. One theoretically correctly identifies dandelions 93% of the time, although a little confused with the tulips and daisies. Roses have a high prediction accuracy, placed at 96%, although a little confused from sunflowers. The prediction to sunflowers is 96%, although there is minimal misclassification with roses and daisies. The tulip has the accuracy of 95%, although it has some of the roses misclassifying. From the metrics here presented, the model is performing very well, especially the dandelions and sunflowers, such that these are highly in precision and F1-scores, respectively.

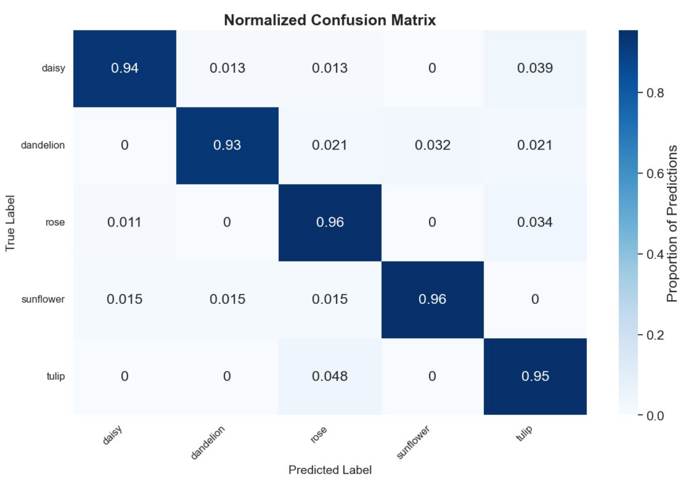

Fig. 13: Normalized confusion matrix to visulize the performance of the flower recognition model.

IV. Hyperparameter Tuning

Hyperparameter tuning is, one of the most critical processes in the development and optimization of neural network models, including CNNs applied to image recognition tasks such as flower classification. In this manner, hyperparameters themselves are just the tunable parameters that change the way models train, hence determining the nature of the learning behavior and generalization from the training dataset.

Model parameters are learned in the learning process, but the set before hand hyperparameters include learning rate, batch size, number of epochs, and choice of the loss function.

We are putting much emphasis on hyperparameter tuning because the correct hyperparameter setting will turn out with a dramatic effect on the result. For hyperparameters set well, the model will go towards better accuracy, faster convergence, and the generation of other data. Poor models are overfit and underfit bad choices, respectively. Optimizing and exploring hyperparameters are crucial for developing robust and high-performance models. This chapter reports the impact of two specific areas of hyperparameter tuning applied to our model of recognition of flowers. The first one deals with checking how performance might change in the absence of applying data augmentation techniques.

A. The Effect of Data Augmentation

Data augmentation artificially expands the training dataset by applying random transformations to enable the model to generalize better exposed to a variety of scenarios possible. When this model is put through a new set of data to be predicted, it most probably will overfit the problem, and the issue in it will lead to lower performance.

Given graph in the Fig 14, compares our convolutional neural network's performance, with and without data augmentation, for the recognition of flowers based on 30 epochs.

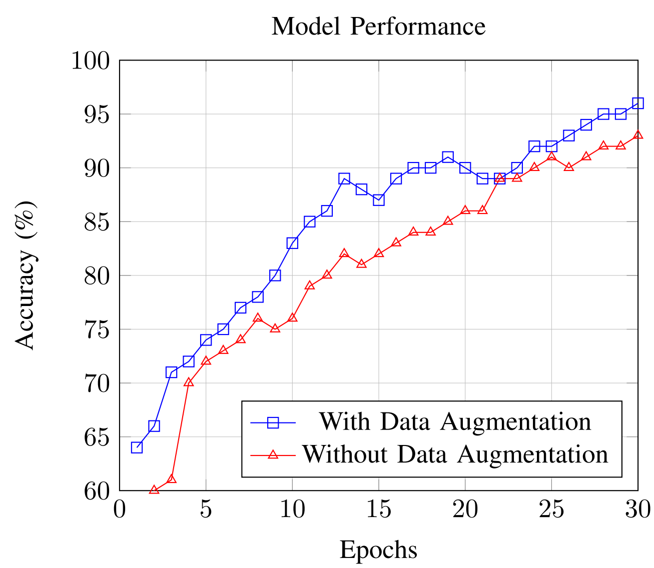

Fig. 14: Comparison of model performance with and without data augmentation over 30 epochs.

The plot demonstrates a rather fine increase in model accuracy after data augmentation procedures. It started at about 65% and linearly increased to 97% at the 30th epoch. For the non-augmented data, its accuracy grew from the same point but with a decreasing trend. By the 30th epoch, it was somewhere around 94%, which indicated that this model was still in the learning process, though much less effective.

The model learns more consistently and rapidly with data augmentation. Hence, data augmentation increases the power of the model to learn in a consistent, rapid manner with the training data. Accuracies fluctuate lesser, thus making learning nature smooth. Learning without data augmentation, on the one hand, is rough in nature and shows a significantly higher amount of fluctuation. Still, on the other hand, it does show a risk of overfitting or being highly sensitive to the model learning from the training data. The Fig. 14 evidences that much better results, from the beginning, happen with the models trained by data augmentation than with their counterparts, which are trained without data augmentation. This translates to data augmentation supporting the model to be better with generalization properties even when seen on a minimal training scale. Theoretically, the primary training difference occurs in the final epochs, so the models trained with data augmentation superseded the models without it in the final accuracy. Data augmentation makes models better.

The dataset used in this study comprises five classes, each containing 1,000 images, ultimately leading to 5,000 photos. It is a good amount of images for training a robust model. Data augmentation would thus be particularly beneficial when the dataset is small or imbalanced non applicable in both cases. Our set of images is good because it contains many different angles, different kinds of illumination, and backgrounds. This intrinsic dataset variability would mimic the effects of data augmentation, and the model would be good at generalization without adding more data augmentations.

While it has been built for the low latency computation for MobileNetV2 architecture, it works better even with smaller datasets than other deep networks. How the architecture of MobileNetV2 was done might, by the end of the day, make the said optimization extraction available even without undergoing data augmentation. We used the pre-trained model MobileNetV2, which was fine-tuned on our flower dataset. Since the pre-trained model has been trained with a wide variety of them in ImageNet which is a large and diversified database, most of the features have been captured in the pre-trained model. Thus, data augmentation is not essential in our case.

Data augmentation induces an overhead in computation since the transformations are on the fly for each image. Very high training times would have been incurred since we would be considering a lot of data augmentation hence, we did not use it. It will be efficient in this way and reach a very high level of accuracy. All in all, in this regard, since the computational resources we were working on were optimized as well, we did not use data augmentation. We mention this to ensure that our training is reasonably restricted by what the hardware can carry out.

It is evident from the Fig. 14 that our model, by the 30th epoch, was approximately 94% accurate without any data augmentation. The 94% performance level, which is high, proved that the model learned and generalized well from the given dataset. This time, the accuracy difference between these two models was slight, only 3%. So we accepted this trade-off, given the benefits provided by the remaining.

Though data augmentation is one of the most powerful strategies in performance improvements, we did not try to reach the limit of its exploitation to the full possibilities of the already large and quality-ample dataset, as well as the efficiency of the MobileNetV2 architecture and the cost-effectiveness in the use of training time and computational resources. Very high accuracy, in combination with the exclusion of data augmentation, serves as yet another reason for confidence in the robustness of our strategy and complete sufficiency of the dataset in providing all necessary variability needed for practical training of the model.

B. The Effect of Loss function

We will also consider how these functions affect them during the training process. The loss function thus evaluates the difference between model predictions and actual outcomes, driving the optimization process. Changing this loss function could potentially impact the learning dynamics of a model and, in that respect, its final performance. An experiment with loss functions can provide us with much insight into how the choice of hyperparameters influences the accuracy and robustness of the CNN model when used to recognize flowers. These are some considerations whereby the worth of such hyperparameters in the analysis will be drawn out and how their optimization helps in the global effectivity of the CNN model when flower recognition tasks are considered.

In the project, the effect of different loss functions on the model's performance in classifying flowers are evaluated. Some of the loss functions applied are Categorical Cross-Entropy, Mean Square Error, and Sparse Categorical Cross-Entropy. The results of these evaluations are then summarized in Table III, giving the precision, recall, and F1-score per each class of flowers, along with the overall accuracy, for each of the loss functions.

Overall, Categorical Cross-Entropy showed the best results in most aspects, being high in accuracy and in balanced metrics among all the classes. Also, the Mean Square Error was high in precision, but it experienced a slight trade-off in recall for the class Sunflower. The least effective was the application of Sparse Categorical Cross-Entropy, which is not the right choice for this specific multi-class classification task. Thus, the results presented in this paper revealed that choosing the appropriate loss function lies at the base of the classification of a flower using the CNN model. Again, the best-performing loss function was categorical cross-entropy, quickly owing to handling the multi-class classification challenges that might be taking place better in this task. This is consequential in actual application in obtaining the desired results through optimal performance and enhanced accuracy in the identification of flowers.

TABLE III: Comparison of classification results using Categorical Cross-Entropy and Mean Square Error loss functions. Precision, Recall, and F1-Score are listed for each class, along with overall accuracy.

Class	**Categorical Cross-Entropy**			**Mean Square Error**			**Sparse Categorical Crossentropy**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Daisy	0.93	0.90	0.92	0.96	0.91	0.93	0.84	0.85	0.87
Dandelion	0.93	0.95	0.94	0.93	0.96	0.94	0.82	0.83	0.86
Rose	0.94	0.87	0.91	0.83	0.90	0.86	0.82	0.88	0.86
Sunflower	1.00	0.94	0.97	0.96	0.89	0.93	0.86	0.87	0.89
Tulip	0.88	0.97	0.92	0.85	0.85	0.85	0.84	0.86	0.84
Accuracy		0.93			0.91			0.88	

V. CHALLENGES

The task of flower detection through the use of convolutional neural networks has been considered an emerging field with high potential towards applications in agriculture and marketing. But in spite of the effort, many challenges are yet to be mitigated in order to reap complete benefits. These come broadly from issues in data collection and cleansing, model implementation, and model accuracy enhancement.

A. Data Collection and Cleaning

However, one of the main issues with the development of a CNN model for the detection of flowers concerns the collection and preparation of high-quality data. CNNs are deep learning models that require large annotated image datasets to learn from, and they are reputedly data-hungry. Datasets tend to be very hard to acquire in terms of human labor and cost. That is why all images should be properly labeled, containing information about the type of flower and other relevant data. Poor or inconsistent labeling of images leads to massive degradation of model performance.

There are various flower images that create diversity in the model in a way that includes lights, angles, backgrounds, and seasons. Flowers look completely different with varying light conditions, and even the same flower will look quite dissimilar when taken at dawn, at noon, at dusk, or at night. The same is true for flowers captured at different angles or on various backgrounds, which could confuse the model otherwise.

Another important challenge is data cleansing. For instance, some raw data contain noise, like some blurry pictures of flowers or obstructed flowers with other objects, and so forth. If given as input to the CNN model, this may cause misleading detections and lead to poor performance. Data cleaning handles filtering noisy data, mishandling of labeling failures in the images, and making uniform quality in the images. It generally requires automated tools and manual inspection in order to have a clean picture dataset, so it is time- and labor-consuming.

B. The Implementation of the CNN Model

Running an implementation of CNNs for flower recognition brings some challenges. The networks are computationally expensive and need high processing power with substantial memory. Convolutional neural network training involves many parameters and hyperparameters, which have to be optimized; therefore, this may become a complex and lengthy process.

It is important to work with the relevant architecture of the previously mentioned neural network: ResNet, VGG, or Inception. Afterward, its hyperparameters can be tuned in accordance, such as learning rate, batch size, and number of epochs, but this requires heavy experimentation and domain knowledge.

Another issue is overfitting, in which the model will work overwhelming well when their training data points are presented and very poorly when new unseen test data points are encountered. Overfitting means that the model has learned by heart from their training data, not general principles.

This is highly problematic in the case of training for flower detection, wherein data set variations may finally contribute to bias. These problems are generally cured with techniques like dropout or data augmentation and regularization, but all that leads to much more enormous model complexity once implemented.

Added to this are the extra challenges of embedding CNNs models into real-world systems, such as agricultural machinery or mobile applications. Most of these systems have very limited computational resources, so it becomes necessary to develop really efficient models that work in real time; for instance, making lightweight models or applying model compression techniques like pruning and quantization. Thus, a major concern in the implementation of these is the compromise between model accuracy and computational efficiency.

C. Enhancing Model Accuracy

Major sources of variation within this problem, which tend to result in quite poor accuracies, are occlusion of flowers by leaves or stems, and changes in appearance due to factors like weather or pests/diseases. Turned around, this source variation poses a challenge to the CNN in identifying flowers consistently.

Sometimes, the prediction score of the model can be further improved by careful finetuning of the model architecture and its parameters. Prediction scores for careful finetuning of the model architecture and parameters are sometimes further increased. Pre-training techniques of a model without any specific task for flower detection improve results. Techniques of transfer learning from big pre-trained models derived while training on the ImageNet dataset have good initialization characteristics, which mostly lead to faster convergence for such datasets and improve their accuracy.

Another way that can boost the model's accuracy is through ensemble methods, where models are trained multiple times and then averages of these predictions are taken. Ensembles create robustness. They approximate the values with more accuracy and are able to average out the errors from its constituent models. This, however, gets even more computationally complex, which might not be suitable for most real-time applications due to this resource-intensive nature by ensembles.

Another domain in which data augmentation becomes very relevant for model performance improvement is that, if the dataset had been artificially enlarged — for example, through rotation, scaling, or flipping of images — the model might become robust to variations on the dataset. A careful choice of strategies for the augmentation is usually afforded. Hyperparameter tuning is another way through which you can further optimize the model's performance by methodically searching a set of hyperparameters to get the optimal configuration. Methods include grid search, random search, and Bayesian methods, which are costly in computation and time.

D. Pragmatic Implementation and Adaptation Challenges

Even more challenging is the adaptation of CNN-based models for such purposes as flower detection into practical use namely, real-time operations of agricultural machinery or marketing platforms where the decision systems are used. The models have to be ruggedized and adaptable because in agriculture, machines are designed to work in very divergent environmental conditions. The efficient algorithms have to be developed to run on the limited hardware of autonomous farming equipment for high real-time processing.

Huge amounts of image data will be dealt with in marketing when flower detection technology is integrated into the e-commerce site or social media for a smooth user experience. The system needs to process quickly in order to analyze and process the input images so that real-time recommendations and personalization based on dynamic content can be implemented with the same performance but high in model accuracy, scalability, and efficiency in deployment. Finally, for both the agricultural and marketing purposes of the produce, there is a perpetual need for learning and adaptation. With new kinds of flowers or changes in consumer preferences, the models must be updated and then retrained, which again demands a solid pipeline for new data to be retrained and deployed updates without a hiccup in services run.

It is very promising yet challenging to apply convolutional neural networks to detect the flower. Successful deployment of such a system deals with the issues related to the data collected and cleaning process, model implementation, and enhancement of model accuracy. The challenges need advanced techniques, with great experimentation and practical considerations needed from real-world applications so as to conceptualize them. Only research and development would go a long way toward ensuring these challenges are addressed so that the application of CNN-based flower detection in the fields becomes more effective and widespread. These challenges can be clearly observed in the Fig. 15.

VI. Future Work

Future research and development in the application of CNNs toward the detection of flowers should hold great promise for such an advancement and future applications. One such focus area could be the extension of the number of flower types to include as many as possible in order to improve system versatility while applied under varied agricultural settings and different floriculture markets. This can be done by curating and annotating a large dataset covering multiple species of flowers, which can then be learned by the CNN model for its identification and differentiation with high accuracy. The less glamorous but probably significantly more currently practical use is the application of the developed flower detection technology as part of the agricultural machinery.

This can optimize the entire agricultural process by incorporating, into platforms of autonomous equipment, CNN-based flower detection systems for weeding robots, sprayers, and harvesters.

Such machines could hence travel a field independent of an operator, identify and respond to distinguishable flowers, and adjust operations as required, thereby increasing the precision of treatments such as targeted weeding, pest control, and selective harvesting. Implementation of such technology would need rugged hardware for real-time processing and an interface between the detection system and the operational controls of the machinery that is seamless.

A couple of improvements upon them can further be done to make the model in the detection of flowers even stronger. Another field which researchers should pay gaze at is the application of techniques that make the model more robust to diverse scenarios of environmental conditions, lighting, and weather. This might be achieved by techniques like data augmentation, transfer learning, and ensemble methods. Further, another area that is enhancing computational resources for deployment on low power devices that are used in the field is useful.

The marketing sector has even more to benefit from this technology. On the other hand, real-time detection of flowers offers the marketer an ability to customize advertisement and product offering according to current trends and consumer choices. For instance, e-commerce sites can utilize CNNs to suggest trending flower bouquets for a rich shopping experience. Social media applications can use flower detection to personalize their content: users will see posts and advertisements that include pictures of their favorite flowers, which will greatly enhance user engagement and conversion rates.

Further research can be carried out to integrate flower detection with AR/VR in developing marketing experiences. This will help the customer have a broadly perceptible view regarding floral arrangements within their environment prior to an actual purchase, hence being able to have greater satisfaction and therefore increase revenues. Real-time data

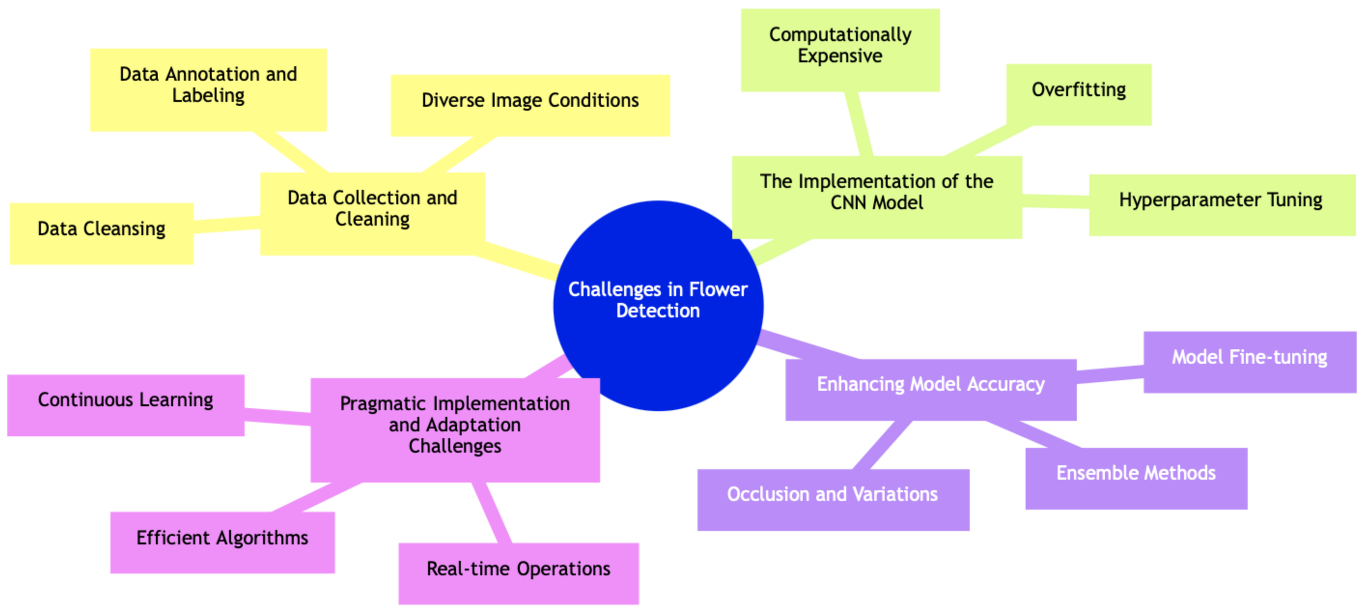

Fig. 15: Diagram illustrates the hierarchical structure of the challenges in using convolutional neural networks (CNNs) for flower detection.

analytics linked with adaptive marketing strategies, based on the insights from flower detection, will make a company agile and competitive in a dynamically changing market landscape. In general, further development and application of CNN-based flower detection have the potential to revolutionize the agricultural as well as marketing sectors by promoting efficiency, precision, and satisfaction.

Conclusion

Implementation of CNN for recognition of flowers already shows a good chance of changing many industries, among which are agriculture and marketing. Using a well-curated dataset and the MobileNetV2 model, our project has shown how well the CNN architecture can be employed in fine-classification problems. The high accuracy of our experiments proves the robustness of CNNs in complex image recognition tasks, which are very important in agriculture, because precision in the identification and classification of flowers can be a game-changer for a farmer. By employing the CNN-based flower detector, the farmer can control targeted weed infestation, analyze the health of the crop, and plan harvesting. The technology hence reduces the amount of manual work, saves time, and boosts the harvest received, making farming more sustainable and profitable. What's more, the integration of CNNs within other farm equipment, such as autonomous weeding robots and precision sprayers, is sure to give even greater precision and effectiveness to these operations with real-time adjustments based on accurate flower detection.

Second, with CNN-based flower recognition, the marketing sector has what it takes. Businesses can produce the latest and most relevant marketing strategies based on the real-time changes in consumer behavior. For instance, the e-commerce platforms can, in real time, suggest the most popular types of flower bouquets based on the real-time detection of consumer preferences, which in turn will make the related shopping experience and sales better. In such a way, social media platforms can give recommendations of personalized content, given that it has detected the favorite flowers of the user on posts and ads, which, in turn, will give more engagement and a higher conversion rate.

However, using CNNs for flower recognition comes with some complications: Data collection and cleaning are important steps and quite laborious to guarantee the quality of the training data. The brightness, angles, and background of flower images are so variable that the model's accuracy is challenged, and only sophisticated data augmentation techniques might make such models general. It is necessary to design efficient models that can work on the limited hardware and associated computational resources of CNN training and deployment, particularly in real time. Further research could really find its opportunities in scaling datasets of the different flower species for an improvement in model versatility and robustness. Transfer learning and ensemble methods will further boost model performance through the use of pretrained models and multiple predictions. With these CNN models increasing in resolution and frequency, there will be a need for advancement in hardware, increase in computational resources, and feasibility for real-world applications.

There is still a huge potential that the CNN-based systems for flower detection could have in the furtherance of the field of precision agriculture and, similarly, in targeted marketing. It is a technology that indeed holds great potential and further opens gates to new opportunities in developing more efficient, sustainable, and profitable practices in these sectors, breaking through challenges and innovating at the same time. The coming together of artificial intelligence with real-time practical application in flower recognition is opening a new era for the growth of tech, heralding exciting future research and development.

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
VGG	Visual Geometry Group
ResNet	Residual Network
YOLO	You Only Look Once
AR/VR	Augmented Reality/Virtual Reality
EDA	Exploratory Data Analysis
mAP	Mean Average Precision
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
IT	Information Technology

References

[1] Subash. S. I, Muthiah. M. A., and N. Mathan. A novel and efficient cbir using cnn for flowers. In *2023 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 1–6, 2023.

[2] Hitesh Kumar Sharma, Tanupriya Choudhury, Rishi Madan, Roohi Sille, and Hussain Mahdi. Deep learning based model for multi-classification of flower images. In Vijay Singh Rathore, Vincenzo Piuri, Rosalina Babo, and Marta Campos Ferreira, editors, *Emerging Trends in Expert Applications and Security*, pages 393–402, Singapore, 2023. Springer Nature Singapore.

[3] Mastura Hanafiah, Mohd Azraei Adnan, Shuzlina Abdul-Rahman, Sofianita Mutalib, Ariff Md Ab Malik, and Mohd Razif Shamsuddin. Flower recognition using deep convolutional neural networks. *IOP Conference Series: Earth and Environmental Science*, 1019(1):012021, apr 2022.

[4] Büşra Rümeysa Mete and Tolga Ensari. Flower classification with deep cnn and machine learning algorithms. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–5, 2019.

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

[6] Chhaya Narvekar and Madhuri Rao. Flower classification using cnn and transfer learning in cnn- agriculture perspective. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 660–664, 2020.

[7] Gao Yifei, Qiu Chuxian, Xu Jiexiang, Miao Yixuan, and Teoh Teik Toe. Flower image classification based on improved convolutional neural network. In *2022 12th International Conference on Information Technology in Medicine and Education (ITME)*, pages 81–87, 2022.

[8] Gandhinee Rajkomar and Sameerchand Pudaruth. A mobile app for the identification of flowers using deep learning. *International Journal of Advanced Computer Science and Applications*, 14(5), 2023.

[9] Johanna Ärje, Dimitrios Milioris, Dat Thanh Tran, Jane Uhd Jepsen, Jenni Raitoharju, M. Gabbouj, Alexandros Iosifidis, and Toke Thomas Høye. Automatic flower detection and classification system using a light-weight convolutional neural network. 2019.

[10] Charu Shree and Rupinder Kaur. A survey on flower detection techniques based on deep neural networking. *International Journal of Engineering Research & Technology (IJERT)*, 8(05):–, 2019.

[11] Salik Ram Khanal, Ranjan Sapkota, Dawood Ahmed, Uddhav Bhattarai, and Manoj Karkee. Machine vision system for early-stage apple flowers and flower clusters detection for precision thinning and pollination. *IFAC-PapersOnLine*, 56(2):8914–8919, 2023.

[12] R Raja Subramanian, Narla Venkata Anand Sai Kumar, Nallamekala Syam Sundar, Nali Harsha Vardhan, Maram Uma Maheshwar Reddy, and M V Sanjay Kumar Reddy. Flowerbot: A deep learning aided robotic process to detect and pluck flowers. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pages 1153–1157, 2022.

[13] Alexander Mamaev. Flowers recognition, 2023.

[14] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[15] Maria-Elena Nilsback and Andrew Zisserman. Delving into the whorl of flower segmentation. In *British Machine Vision Conference*, volume 1, pages 570–579, 2007.

Authors

Burak Erdil Biçer is currently pursuing a double major in Electronics and Communications Engineering and Aeronautical Engineering at Istanbul Technical University.

He interned at Baykar Technologies, focusing on computer vision algorithms using OpenCV and TensorFlow, and at Aselsan, where he developed communication interfaces for LoRa E32 modules. As a long-term Software Engineering Intern at Turkish Aerospace, he enhanced his programming skills in C and C++. Burak is also an Undergraduate Research Assistant at the BOUNtenna (Boğaziçi University Antennas Propagation Research Lab), working on biodegradable sensors and phantom design for IoHT applications.

Javad Ibrahimli is currently an Electronics and Communication Engineering student at Istanbul Technical University, with a strong focus on Computer Vision, Machine Learning, and Autonomous Vehicles.

He interned as an Autonomous Systems Engineer at the ITU ZES Solar Car Team, where he developed software modules for autonomous driving systems, focusing on perception, planning, and control. He leveraged his expertise in machine learning, computer vision, and sensor technology to create innovative solutions for autonomous navigation and obstacle detection. Additionally, he configured and calibrated sensors such as LIDAR, cameras, and GPS for accurate data acquisition.

Mehmet Anıl Korkmaz is a senior student at Istanbul Technical University, departmant of Electronics and Communication Engineering.

He has developed his skills in embedded systems architecture and Embedded C programming as a member of the embedded systems group in the ITU Solar Car Team. Currently, he is working part-time embedded software developer at Borda Tech, focusing on the development of Bluetooth-based IoT products.